

$$\text{Var}[x] = \mathbb{E}[(x - \mathbb{E}[x])^2]$$

On the plot below there is a normal (Gaussian) distribution with the same value of mean $\mu = 10$

Now lets return to our policy gradients. It has already been said in the previous chapter, the method idea is to increase the probability of good actions and decrease the chance for bad actions. In math notation our policy gradient was written as $\nabla J \approx \mathbb{E}[Q(s, a)\nabla \log \pi(a|s)]$

1. Initialize network parameters θ with random values
2. Play N steps in the environment using the current policy π_θ , saving state s_t , action a_t , reward r_t
3. $R = 0$ if the end of the episode is reached or $V_\theta(s_t)$
4. For $i = t - 1 \dots t_{start}$
 - (a) $R \leftarrow r_i + \gamma R$
 - (b) Accumulate the policy gradients $\partial\theta_\pi \leftarrow \partial\theta_\pi + \nabla_\theta \log \pi_\theta(a_i|s_i)(R - V_\theta(s_i))$
 - (c) Accumulate the value gradients $\partial\theta_v \leftarrow \partial\theta_v + \frac{\partial(R - V_\theta(s_i))^2}{\partial\theta_v}$
5. Update network parameters using the accumulated gradients, moving in the direction of policy gradients $\partial\theta_\pi$ and in the opposite direction of the value gradients $\partial\theta_v$.
6. Repeat from step 2 until convergence

Entropy bonus is usually added to improve exploration. Its usually written as an entropy value added to the loss function: $\mathcal{L}_H = \beta \sum_i \pi_\theta(s_i) \log \pi_\theta(s_i)$.

Forward pass through the network returns tuple of two tensors: policy and value. Now we have large and important function which takes the batch of environment transitions and returns three tensors: batch of states, batch of actions taken and batch of Q-values calculated using the formula $Q(s, a) = \sum_{i=0}^{N-1} \gamma^i r_i + \gamma^N V(s_N)$

The last piece of our loss function is entropy loss which equals to the scaled entropy of our policy taken with the opposite sign (entropy is calculated as $H(\pi) = - \sum \pi \log \pi$)